Iranian Society of Acoustics and Vibration

The 13th International Conference on Acoustics and Vibration ISAV2023

20 , 21 Dec 2023 Tehran - Iran

Leveraging Koopman Operator and Deep Neural Networks for Parameter Estimation and Future Prediction of Duffing Oscillators

Yassin Riyazi [a],NavidReza Ghanbari [a], Arash Bahrami [a*]

[a] *M.Sc. Student, School of Mechanical Engineering, College of Engineering, University of Tehran, Tehran, Iran.*

[b] *Assistant Professor, School of Mechanical Engineering, College of Engineering, University of Tehran, Tehran, Iran.*

** Corresponding author e-mail: arash.bahrami@ut.ac.ir (A. Bahrami)*

Abstract

The study of nonlinear dynamical systems has been a cornerstone in various scientific and engineering fields due to their widespread applications in modeling real-world phenomena. Traditional methods for analyzing and predicting the behavior of such systems often involve complex mathematical techniques and numerical simulations. This paper introduces a novel approach that combines the power of the Koopman Operator and deep neural networks to generate a linear representation of the Duffing oscillator, enabling effective parameter estimation and accurate prediction of its future behavior. Furthermore, a modified loss function is proposed to enhance the training process of the deep neural network. The synergy of the Koopman Operator and deep neural networks not only simplifies the analysis of nonlinear systems but also opens a promising avenue for advancing predictive modeling in various fields.

Keywords: Koopman Operator; Parameter Estimation; Duffing oscillator; deep neural networks; nonlinear dynamical systems; predictive modeling.

1. Introduction

Nonlinear dynamical systems, recognized for their intricate and sometimes chaotic behavior, permeate the realms of natural phenomena and technological applications. They transcend the simplicity of linear systems, giving rise to phenomena such as bifurcations, limit cycles, and chaotic attractors. These systems have long captivated the interest of scientists and engineers, presenting substantial challenges in understanding, characterizing, and predicting their trajectories. Across diverse fields, from physics and biology to economics and engineering, nonlinear systems underscore the fundamental complexity of our world.

At the heart of this intricate landscape lies the Duffing Oscillator (1), an iconic archetype of nonlinear dynamical systems. Its versatility enables it to emulate a wide spectrum of behaviors, making it a pertinent model for various physical phenomena. From capturing the subtle interplay of mechanical vibrations in structures subjected to external forces to mirroring the rhythmic patterns of biological oscillations, the Duffing oscillator encapsulates the essence of nonlinear dynamics.

Traditionally, dissecting and forecasting the behavior of Duffing oscillators have relied on a combination of analytical techniques and numerical simulations. While these methods provide valuable insights, they often encounter limitations in handling nonlinear intricacies with precision. Analytical solutions may prove elusive or algebraically expensive, especially for higher-dimensional or strongly nonlinear systems. On the contrary, numerical simulations, though powerful, demand extensive computational resources and face challenges in long-term predictions due to inherent numerical errors and uncertainties.

The Koopman Operator (2), with its inherent structure involving a mapping to a higher dimension, a linear transformation, and an inverse mapping, bears resemblance to the structure of autoencoders (3). Acknowledging the pioneering work of S. L. Brunton and J. N. Kutz, who introduced the SINDY method (4), we appreciate the potential for leveraging the generality offered by deep learning to successfully identify systems (5,6).

While the sole use of a neural network proves accurate and meets our requirements, it does not guarantee the exclusive confinement of the Koopman Operator to a designated linear layer. Consequently, the entire network structure incorporates elements of mapping, linear transformation, and inverse mapping simultaneously. This challenges the utility of using the linear layer weights as a representation of the system, as they only encapsulate a portion of the Koopman Operator.

To overcome these challenges, we introduce an innovative approach that capitalizes on the synergy between the Koopman Operator and Deep Neural Networks (7–9). This groundbreaking fusion is aimed at converting the Duffing oscillator into a linearized representation, offering promising solutions to the intricacies encountered in traditional methods. By harnessing the computational power of deep learning and the Koopman operator's capability to provide a linear representation of nonlinear systems (10), our approach enables a more accurate Koopman linearized representation of system behavior.

Our approach not only streamlines the analysis of nonlinear systems but also extends its applicability across a diverse spectrum of domains. It ushers in a new era in predictive modeling by opening doors to effective parameter estimation and precise future predictions, addressing the challenges posed by the inherent complexity of nonlinear systems.

In the following sections, we dig into the foundational principles of Koopman Operator theory and the adaptability of deep neural networks. We show how their fusion forms a compelling framework for analyzing and predicting the behavior of Duffing oscillators. We outline the process of transforming Duffing oscillator dynamics into a linear representation and introduce a modified loss function designed to enhance the generality of the Koopman linear representation of the dynamical system within this context. Through numerical validation and comparisons with traditional methods, we demonstrate the efficacy of our approach in providing accurate predictions for the future behavior of Duffing oscillators. Ultimately, this work enriches our understanding of nonlinear dynamics and offers a powerful tool with transformative potential across scientific, engineering, and practical applications.

2. Koopman Operator and Its Application

The exploration of dynamical systems has long been a cornerstone in understanding complex behaviors in various scientific and engineering disciplines. Traditionally, the analysis of these systems has been deeply entwined with the concept of state space, where the evolution of a system is

represented by trajectories in the space of its state variables. However, the inherent nonlinearity of many real-world systems often makes their analysis and prediction challenging using conventional techniques.

In recent years, the Koopman Operator has emerged as a powerful mathematical tool that provides a fresh vantage point for studying dynamical systems. Rooted in functional analysis, the Koopman Operator introduces a paradigm shift by transitioning the focus from the state space to the space of observable functions. Doing so allows us to view the system's evolution in a linear framework, even when dealing with inherently nonlinear systems. This perspective offers a new lens through which we can gain insights into the dynamics of complex systems.

2.1 Dynamical System Representation

Consider a dynamical system described by a set of state variables $\mathbf{x}(t)$, which evolve over time t. Mathematically, we can represent this as:

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}, t) \tag{1}$$

where

- n is the state vector dimension.
- $\mathbf{x} \in \mathbb{R}^n$ is the state vector representing the system's state variables.
- $t \in \mathbb{R}^+$ is time.
- $f(\mathbf{x}): \mathbb{R}^n \to \mathbb{R}^n$ is a vector-valued function describing how the state variables change over time.

2.2 Koopman Operator Transformation

The Koopman operator, denoted as $\mathcal{K}$, is an infinite-dimensional linear operator that acts on observables or functions of the state variables. Let $g(x)$ be such an observable. The Koopman Operator maps this observable from the state space to a higher-dimensional space:

$$\mathcal{K}g(\mathbf{x}) = g\left(f(\mathbf{x})\right) \tag{2}$$

Where:

- m supposed to be infinite-dimensional but in numerical approximation a value will be assigned.
- $\mathcal{K}(.): \mathbb{C}^m \to \mathbb{C}^m$ is the Koopman Operator Generator.
- $g(\mathbf{x}): \mathbb{R}^n \to \mathbb{C}^m$ is an observable or function defined on the state space.
- $g\left(f(\mathbf{x})\right)$ represents the observable after the system evolves according to $f(\mathbf{x})$.

2.3 Koopman Operator in Discrete Time

In discrete-time dynamical systems, the Koopman Operator is applied at discrete time steps. For these systems Equation (2) may be represented as:

$$\mathcal{K}g(\mathbf{x}_k) = g(\mathbf{x}_{k+1}) \tag{3}$$

where:

- $\mathbf{x}_k$ represents the state of the system at time k.
- $\mathbf{x}_{k+1}$ represents the state of the system at the next time step $k+1$.

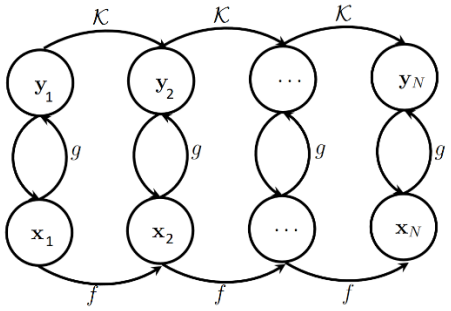

Figure 1. Koopman Operator Evolution and a discrete dynamical system

In Figure 1. an important relationship becomes apparent: the transformation of data x_t from its low-dimensional representation in Euclidean space to an infinite-dimensional Hilbert space is facilitated through the utilization of Koopman observables $g(x_t)$. Leveraging this Koopman transformation allows for the mapping of y_t to y_{t+1} via a linear matrix transformation. Furthermore, employing the inverse Koopman observable mapping $g^{-1}(y_{t+1})$ enables the derivation of x_{t+1}. With the help of the Koopman operator $f(x_t) = g^{-1}(\mathcal{K} \times g(x_t)) = x_{t+1}$.

3. Coupling Koopman Operators with Deep Neural Networks

Deep neural networks have showcased remarkable abilities in approximating intricate functions and mastering complex patterns from data. One of the key challenges encountered in the realm of Koopman Operators is the identification of suitable observable functions. In methods such as DMD (11), the observable function is typically the identity function, and in extended DMD (EDMD), observable functions take the form of polynomials or trigonometric functions. While these approaches are straightforward and accurate, they exhibit resilience to noise and initial conditions.

In this study, we combine a deep neural network with the Koopman operator, thereby generating a linearized representation of the Duffing oscillator. This neural network effectively learns the intricate relationship between system parameters and observed behaviors, facilitating efficient parameter estimation. Moreover, the neural network undergoes training to predict the future trajectory of the Duffing oscillator, thereby equipping us with a valuable tool for forecasting system behavior.

neural networks are not necessarily always better than feature crosses, but neural networks do offer a flexible alternative that works well in many cases.

3.1 Data acquisition

The Duffing oscillator is a dynamical system described by the following second-order differential equation:

$$\frac{d^2x}{dt^2} + \delta\frac{dx}{dt} + \alpha x + \beta x^3 = \gamma\cos(\omega t) \tag{5}$$

Here:

- x represents the displacement of the oscillator from its equilibrium position.

- δ denotes the damping coefficient.
- α is the linear stiffness coefficient.
- β characterizes the nonlinearity in the system.
- γ is the amplitude of the external driving force.
- ω is the angular frequency of the driving force.

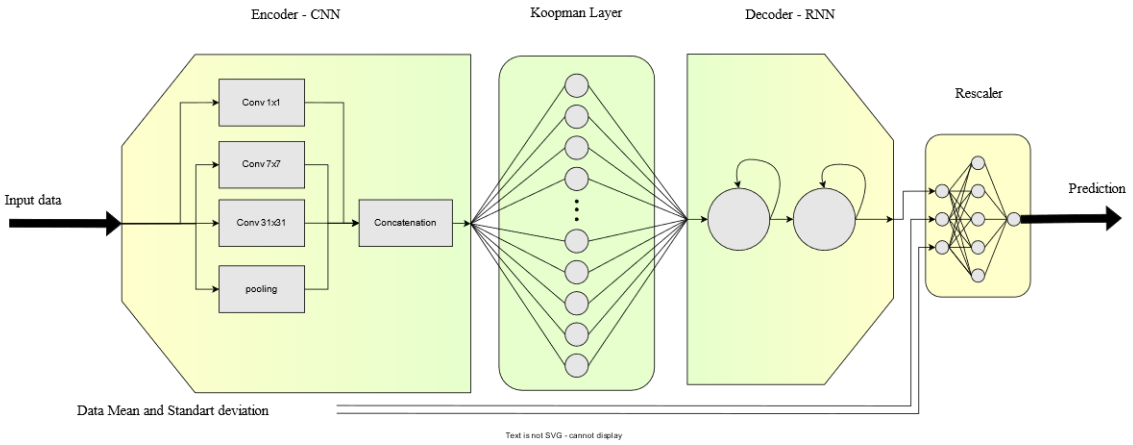

Figure 2. Neural Network Diagram

Duffing oscillator solution has been generated using the Runge-Kuta method (12) and the initial condition for Solving the equation is $x_0 = 1.5$ and $v_0 = -1.5$.

A normal distribution with noise in the range of [-0.5,0.5] was added to the data to simulate real-world data.

The elucidation of the numerical solution has been provided within the Appendix dedicated to Numerical simulation.

3.2 Data Normalization

Given the nature of regression, it is advisable to normalize the data before training the neural network. This normalization is crucial because even small variations in the input data can lead to significant changes in the output, potentially reducing the model's robustness against changes in input conditions. In our approach, we employ data normalization before feeding it into the convolutional neural network (CNN) architecture. During this process, we pass the statistical properties of the data, such as its mean and variance, through the network.

In Figure 2. Neural Network , After the recurrent neural network (RNN) block, the data is re-mapped to its original statistical properties before being passed to the Rescaler Block. The purpose of the Rescaler Block is to further reduce the variation in the output, ultimately leading to a more stable and controlled model response.

3.3 Structure

The architecture of the model as shown in Figure 2. Neural Network is structured around an Encoder-Decoder paradigm, which effectively captures the essence of complex dynamics. Specifically, the Encoder component is meticulously designed, featuring a sequence of Inception Blocks (13) in a convolutional neural network (CNN) (14). These Inception Blocks serve as robust feature extractors, enabling the model to discern intricate patterns and relevant features from the input data.

Following the Inception-based Encoder, a pivotal transformation takes place through a linear layer. This linear layer assumes a distinct role within the architecture, embodying the essence of the

Koopman Operator evolution function $\mathcal{K}$. It is important to note that this linear layer operates without an activation function and bias, preserving the linear nature of the Koopman operator's transformation.

Transitioning from the Koopman Operator layer, the architecture takes an intriguing turn with the integration of a two-layer Long Short-Term Memory (LSTM) (15) network. This LSTM component acts as the Decoder, expertly leveraging its sequential memory to unravel the transformed linearized representation. This sequence-to-sequence modeling approach facilitates the reconstruction of the system's temporal evolution, a crucial aspect in capturing its intricate behaviors.

Table 1. CNN parameters. Out hyperparameter is 20

	Sequential Blocks	In Channels	Out Channels	kernel size	padding
branch1x1	Conv 1×1	1	out	1	0
	ReLU	-	-	-	-
branch7x7	Conv 1×1	1	out	1	0
	ReLU	-	-	-	-
	Conv 7×7	out	out	7	3
	ReLU	-	-	-	-
branch31x31	Conv 1×1	1	out	1	0
	ReLU	-	-	-	-
	Conv 31×31	out	out		15
	ReLU	-	-	-	-
branch pool	MaxPool1d	1	1	3	1
	-	-	-	-	-
	Conv 1×1	1	out	1	0
	ReLU	-	-	-	-

3.4 Training

The network is trained end-to-end, without the need for custom loss functions or specialized training algorithms. However, it is important to note that the evolution function of the Koopman Operator does not remain confined solely to the Koopman part; instead, it spreads throughout the network. In a sense, the network operates as a black box, handling this evolution internally.

To address this issue and restrict the Koopman Operator's influence exclusively to the Koopman linear layer, a two-stage training algorithm has been proposed. In this algorithm, after each optimization step:

1. The weights of all layers except the Koopman Linear Layer are frozen.
2. The output of the Koopman Linear Layer is calculated for time steps n_0 to n_{KPH} (KPH is the Hyper parameters and due to cost of calculating matrix power 20 was selected).
3. The weights of the Koopman Linear Layer are updated based on the linearity property. This update aims to minimize the prediction error of the nth output .

$$\sum_{n}^{KPH} \mathcal{L}\left(g\left(x_{n_0}\right) \times \left(W_{Koopman}^{n}\right)^{T}, g\left(x_{n}\right) \times \left(W_{Koopman}\right)^{T} \right) \tag{4}$$

By implementing this two-stage training process, we ensure that the Koopman Operator's influence is confined and utilized specifically within the Koopman linear layer, enhancing the network's predictive accuracy and control.

Table 2. Optimizer and loss parameters

		Stage 1	Stage 2
Optimizer	Type	SGD	SGD
	Learning Rate	5.00E-02	5.00E-04
	momentum	0.9	-

	weight decay	1.00E-04	-
loss	Type	MSE	MSE

4. Results and Discussion

Numerical results demonstrate the effectiveness of the proposed approach. The combination of Koopman operator-based linearization and deep neural networks yields impressive results in terms of parameter estimation accuracy and future prediction. During the Networks training input horizon for extracting feature was 200 previous sample.

For additional details, please refer to Appendices A and B.

4.1 Simple periodic $\gamma = 0.2$

In the context of simple periodic oscillations, neural networks demonstrate an ability to effectively capture the underlying oscillatory structure, yielding accurate predictions even in the presence of substantial noise.

Figure 3 depicts various scenarios: a) illustrates the neural network's performance under normal training conditions, b) presents a similar scenario with an increased noise level, and c) represents a worst-case situation where noise completely overwhelms the available data, resulting in the network's inability to provide accurate predictions.

In general, neural networks exhibit robustness against noise within a range spanning from $0.0 to \pm 1$.

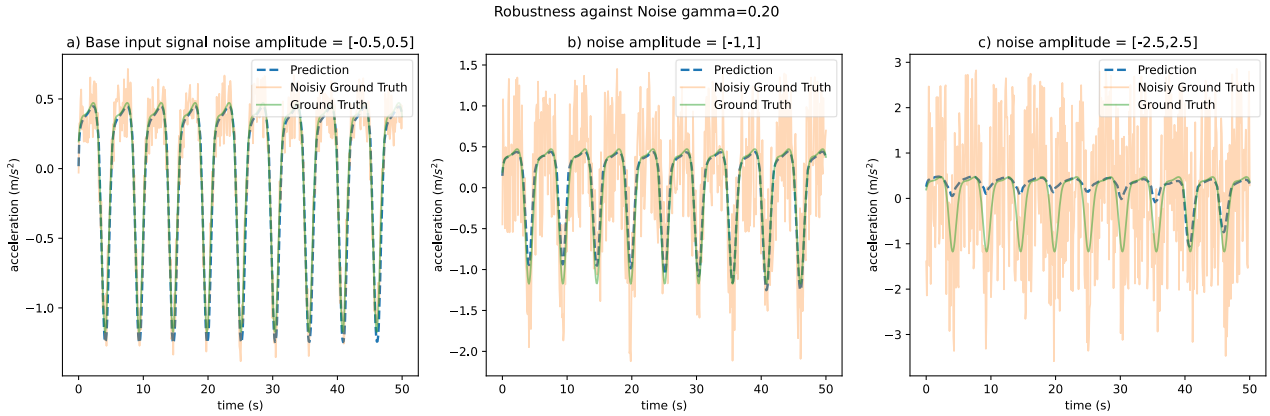

Figure 3. robustness against Noise Variation $\gamma = 0.2$

4.2 Quasi-periodic $\gamma = 0.37$

In the context of quasi-periodic oscillations, akin to simple periodic ones, neural networks exhibit the ability to effectively capture the underlying oscillatory structure, leading to accurate predictions even in the presence of significant noise.

Figure 4 delineates various scenarios: a) showcases the neural network's performance under typical training conditions, b) illustrates a similar scenario but with an elevated noise level, and c) portrays a worst-case scenario where noise completely dominates the available data, resulting in the network's inability to provide precise predictions.

Broadly speaking, neural networks demonstrate robustness against noise within a range spanning from 0.0 to ±1.0."

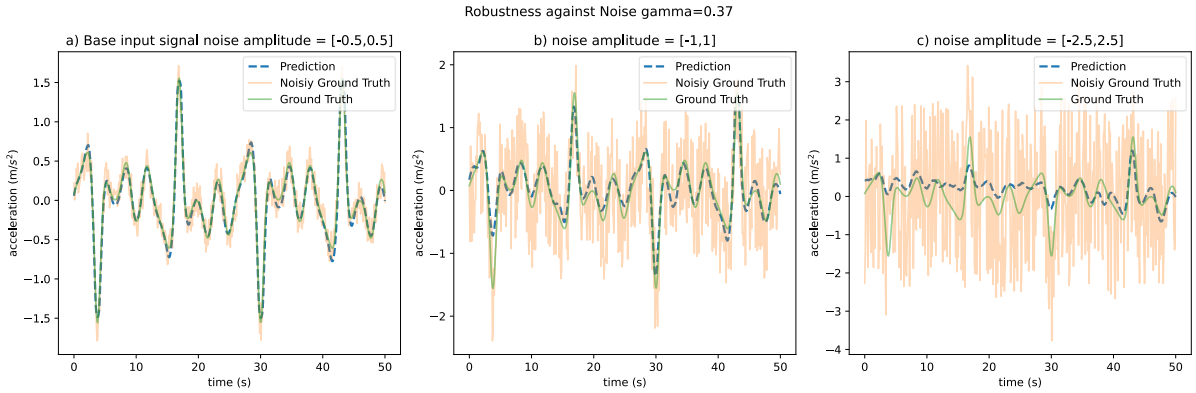

Figure 4. robustness against Initial Noise Variation $\gamma = 0.37$

4.3 Koopman Eigenvalues

One of the drawbacks associated with Dynamic Mode Decomposition (DMD) (11) and Extended Dynamic Mode Decomposition (EDMD) methods (16) is the presence of low Koopman Operator eigenvalues. To address this issue, a radial basis function has been proposed in a previous study (17) to approximate the Koopman Operator eigenvalues.

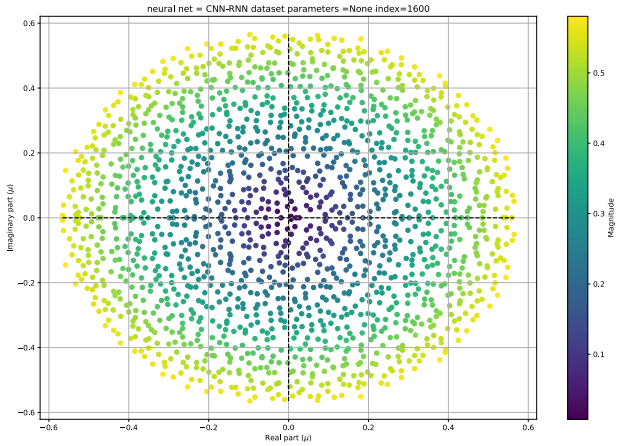

Figure 5. Koopman Layer Eigen values

In Figure 5, we present a plot of 1600 Koopman Operator eigenvalues, demonstrating their diversity and their ability to capture various system behaviors. It is noteworthy that, in the study mentioned [17], these results were obtained for a free Duffing oscillator. Nevertheless, it is evident that the results achieved through Deep Neural Networks surpass those obtained by other means.

5. Conclusion

In a broader context, the amalgamation of the Koopman Operator and Neural Networks reveals substantial potential. The network has proficiently captured the intrinsic data structure, and with further refinements, it holds the capacity to generalize effectively to more intricate problem domains.

The training process was executed as a self-supervised autoregressive procedure conducted in two stages, specifically designed to confine the Koopman Operator's influence to the linear component. It is noteworthy that noise was intentionally introduced into the input data, and the ground truth values utilized for backpropagation error are clearly illustrated in Figure 3 and Figure 4. In essence, the neural networks function as denoisers. Furthermore, the employed data normalization method and the Rescaler block, as depicted in the final block of Figure 2, contribute to this denoising process.

The autoregressive form is instrumental in enabling the network to generate predictions as required, reminiscent of the capabilities exhibited by large language models.

In future investigations, it may prove intriguing to explore the incorporation of Mixture Density Networks (MDNs) for confidence estimation. Particularly in scenarios characterized by chaotic dynamics, the inclusion of MDNs at the model's output holds the potential to provide valuable confidence estimates. This can significantly enhance the reliability of predictions, particularly when dealing with inherently uncertain or complex data.

For access to the project's code and resources, please refer to the GitHub repository available at: github.com/yriyazi/ISAV_2023/

REFERENCES

1. Hamel. Georg Duffing, Ingenieur: Erzwungene Schwingungen bei veränderlicher Eigenfrequenz und ihre technische Bedeutung. Sammlung Vieweg. Heft 41/42, Braunschweig 1918. VI+134 S. ZAMM - J Appl Math Mech / Zeitschrift für Angew Math und Mech. 1921;1(1).
2. Koopman BO. Hamiltonian Systems and Transformation in Hilbert Space. Proc Natl Acad Sci. 1931;17(5).
3. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science (80-). 2006;313(5786).
4. Brunton SL, Proctor JL, Kutz JN. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc Natl Acad Sci U S A. 2016;113(15).
5. Lusch B, Kutz JN, Brunton SL. Deep learning for universal linear embeddings of nonlinear dynamics. Nat Commun. 2018;9(1).
6. Champion K, Lusch B, Nathan Kutz J, Brunton SL. Data-driven discovery of coordinates and governing equations. Proc Natl Acad Sci U S A. 2019;116(45).
7. Krizhevsky A, Sutskever I, Hinton GE. 2012 AlexNet. Adv Neural Inf Process Syst. 2012;
8. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings. 2015.
9. Shafiq M, Gu Z. Deep Residual Learning for Image Recognition: A Survey. Vol. 12, Applied Sciences (Switzerland). 2022.
10. Brunton SL, Budišić M, Kaiser E, Kutz JN. Modern Koopman Theory for Dynamical Systems. SIAM Rev. 2022;64(2).
11. Schmid PJ. Dynamic mode decomposition of numerical and experimental data. J Fluid Mech. 2010;656.
12. Runge C. Ueber die numerische Auflösung von Differentialgleichungen. Math Ann. 1895;46(2).
13. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2015.
14. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE. 1998;86(11).
15. Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Comput. 1997;9(8).
16. Williams MO, Kevrekidis IG, Rowley CW. A Data–Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition. J Nonlinear Sci. 2015;25(6).
17. Li Q, Dietrich F, Bollt EM, Kevrekidis IG. Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the koopman operator. Chaos. 2017;27(10).

Appendix

A) Numerical simulation

The Runge-Kutta method is a numerical technique used for solving ordinary differential equations (ODEs), such as those governing the behavior of the Duffing oscillator. It's commonly employed when analytical solutions are difficult to obtain. Here's an explanation of how the Runge-Kutta method can be applied to solve the Duffing oscillator equation in third person, including mathematical notation:

The Duffing oscillator is described by the second-order ordinary differential equation:

$$m\ddot{x}(t)+k\dot{x}(t)+cx(t)+\alpha x^3(t)=F(t) \tag{a}$$

Where:

- k is the stiffness constant.
- α is a coefficient that determines the strength of nonlinearity.
- $x(t)$ is the displacement of the oscillator at time t.
- c is the damping coefficient.
- $\dot{x}(t)$ represents the first derivative of $x(t)$ with respect to time t, which is velocity.
- m is the mass of the oscillator.
- $\ddot{x}(t)$ represents the second derivative of $x(t)$ with respect to time t, which is acceleration.
- $F(t)$ is the external force applied to the oscillator at time t.

To apply the Runge-Kutta method to solve this equation, by convert this second-order ODE into a system of first-order ODEs. and introducing a new variable, such as $v(t)$, to represent the velocity $\dot{x}(t)$. Then, two first-order ODEs become:

$$\begin{aligned}\dot{x}(t)&=v(t)\\ \dot{v}(t)&=\frac{1}{m}\left[\mathrm{F}(t)-cv(t)-kx(t)-\alpha x^3(t)\right]\end{aligned} \tag{b}$$

To solve this equation using the Runge-Kutta method, the following steps are typically followed:

1. Discretization: Divide the time interval over which you want to solve the equation into small time steps. Let Δt represent the size of each time step, and create a time grid with time points $t_0, t_1, t_2, \ldots, t_n$ where $t_i = t_0 + i\Delta t_i$.
2. Initialization: Set the initial conditions for the displacement and velocity, x_0 and v_0, at t_0.
3. Iteration: For each time step i, perform the following calculations:
   3.1. Calculate the acceleration a_i at time t_i using the Duffing oscillator equation [reference to Duffing equation]
   3.2. Use the Runge-Kutta method to update the displacement and velocity for the next time step:

$$k_1 = \Delta t\left(v_i\right)$$
$$l_1 = \Delta t\left(a_i\right)$$
$$k_2 = \Delta t\left(v_i + \frac{1}{2}l_1\right)$$
$$l_2 = \frac{\Delta t}{m}\left(F\left(t_i + \frac{\Delta t}{2}\right) - c\left(v_i + \frac{l_1}{2}\right) - k\left(x_i + \frac{k_1}{2}\right) - \alpha\left(x_i + \frac{k_1}{2}\right)^3\right)$$
$$k_3 = \Delta t\left(v_i + \frac{1}{2}l_2\right)$$
$$l_3 = \frac{\Delta t}{m}\left(F\left(t_i + \frac{\Delta t}{2}\right) - c\left(v_i + \frac{l_2}{2}\right) - k\left(x_i + \frac{k_2}{2}\right) - \alpha\left(x_i + \frac{k_2}{2}\right)^3\right)$$
$$k_4 = \Delta t\left(v_i + \frac{1}{2}l_3\right)$$
$$l_4 = \frac{\Delta t}{m}\left(F\left(t_i + \frac{\Delta t}{2}\right) - c\left(v_i + \frac{l_3}{2}\right) - k\left(x_i + \frac{k_3}{2}\right) - \alpha\left(x_i + \frac{k_3}{2}\right)^3\right) \quad \text{(c)}$$

3.3. Update the displacement and velocity for the next time step:

$$x_{i+1} = x_i + \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$
$$v_{i+1} = v_i + \frac{1}{6}\left(l_1 + 2l_2 + 2l_3 + l_4\right)$$

4. Repeat step 3 for each time step until you reach the desired endpoint.

The Runge-Kutta method iteratively approximates the solution to the Duffing oscillator equation by considering the rate of change of displacement and velocity at each time step, providing a numerical solution for $x(t)$ and $v(t)$ over the specified time interval.

B) Duffing Solution

Simple Harmonic:

The temporal evolution and phase-space representation of the solution, denoted as $x(t)$, for the Duffing equation are examined. The Duffing equation is Equation 5.

In Figure 6, consider $\gamma = 0.20$ as a representative amplitude. The remaining parameters are set as follows: $\alpha = -1$, $\beta = +1$, $\delta = 0.3$, and $\omega = 1.2$. The initial conditions for the system are $x(0) = 0.5$ and $\dot{x}(0) = 0$.

The temporal evolution is presented as a time series, where x is plotted as a function of $\frac{t}{T}$, with $T = \frac{2\pi}{\omega}$ representing the period of the oscillation. Additionally, the phase portrait is constructed, depicting the time series plotted in the $x - \dot{x}$ phase plane. Notably, the red dots in the phase portrait correspond to instances when t is an integer multiple of the period T.

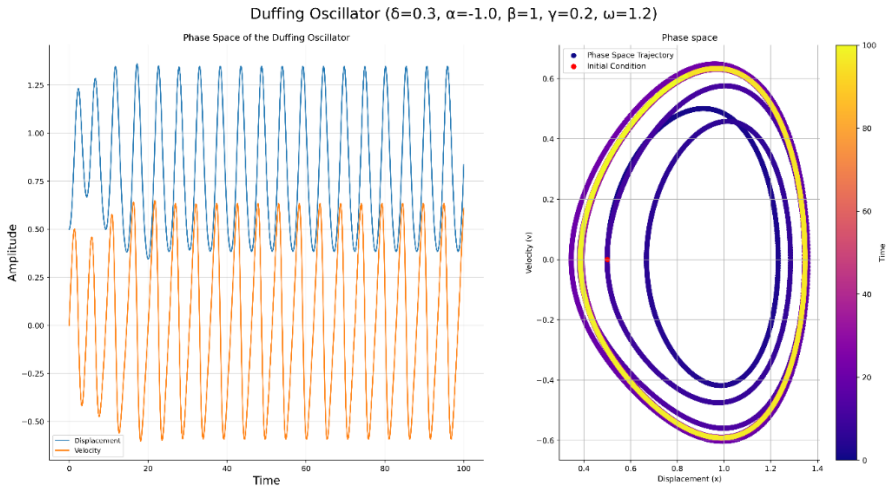

Figure 6. . $\alpha=-1$, $\beta=+1$, $\delta=0.3$, $\gamma=0.20$, and $\omega=1.2$. initial conditions $x(0)=0.5$ and $\dot{x}(0)=0$

Quasi-Periodic Behavior

In, Figure 7. consider $\gamma=0.29$ as a representative amplitude. The remaining parameters are set as follows: $\alpha=-1$, $\beta=+1$, $\delta=0.3$, and $\omega=1.2$. The initial conditions for the system are $x(0)=0.5$ and $\dot{x}(0)=0$.

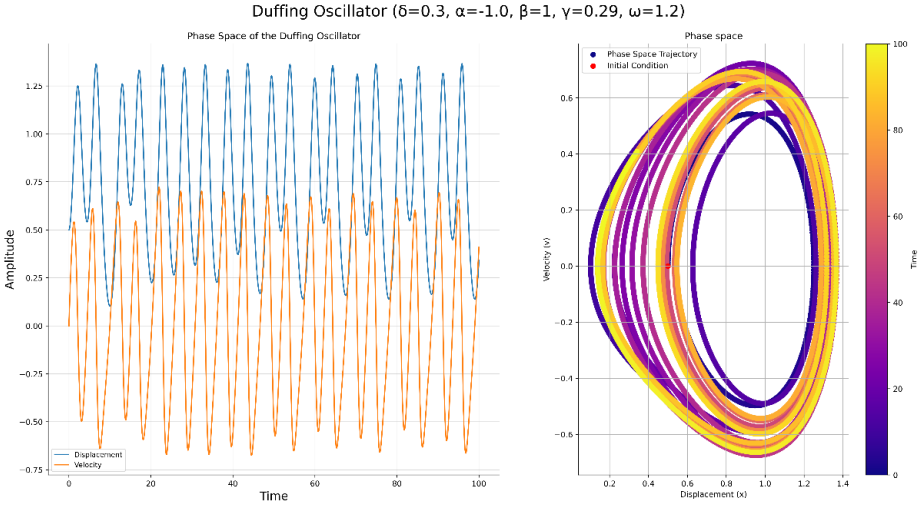

Figure 7. $\alpha=-1$, $\beta=+1$, $\delta=0.3$, $\gamma=0.29$, and $\omega=1.2$. initial conditions $x(0)=0.5$ and $\dot{x}(0)=0$

Quasi-Periodic Behavior

In, Figure 7. consider $\gamma=0.37$ as a representative amplitude. The remaining parameters are set as follows: $\alpha=-1$, $\beta=+1$, $\delta=0.3$, and $\omega=1.2$. The initial conditions for the system are $x(0)=0.5$ and $\dot{x}(0)=0$.

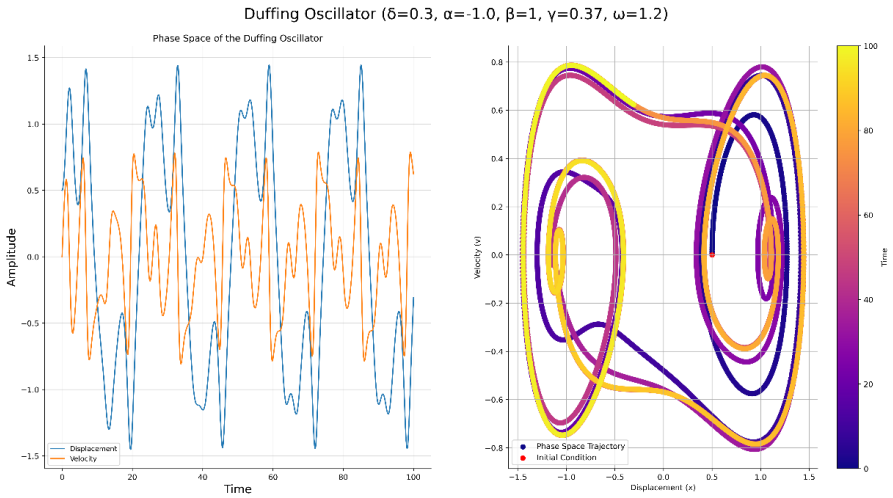

Figure 8. $\alpha=-1$, $\beta=+1$, $\delta=0.3$, $\gamma=0.37$ **, and** $\omega=1.2$ **. . initial conditions** $x(0)=0.5$ **and** $\dot{x}(0)=0$